IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 13, 2021, accepted January 28, 2021, date of publication February 1, 2021, date of current version February 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056137

Two-Branch Convolutional Sparse Representation for Stereo Matching

CHUNBO CHENG[1,2], HONG LI[1], (Member, IEEE), AND LIMING ZHANG[3], (Member, IEEE)
[1]School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, China
[2]College of Science, Hubei Polytechnic University, Huangshi 435000, China
[3]Faculty of Science and Technology, University of Macau, Macau, China

Corresponding author: Hong Li (hongli@hust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61877021, in part by the Research Committee of University of Macau Multi-Year Research under Grant MYRG2018-00111-FST, and in part by the Science and Technology Development Fund (FDCT) of Macao Special Administrative Region (SAR) under Grant 079/2016/A2.

ABSTRACT Supervised learning methods have been used to calculate the stereo matching cost in a lot of literature. These methods need to learn parameters from public datasets with ground truth disparity maps. Due to the heavy workload used to label the ground truth disparities, the available training data are limited, making it difficult to apply these supervised learning methods to practical applications. The two-branch convolutional sparse representation (TCSR) model is proposed in the paper. It learns the convolutional filter bank from stereo image pairs in an unsupervised manner, which reduces the redundancy of the convolution kernels. Based on the TCSR model, an unsupervised stereo matching cost (USMC), which does not rely on the truth ground disparity maps, is designed. A feasible iterative algorithm for the TCSR model is also given and its convergence is proven. Experimental results on four popular data sets and one monocular video clip show that the USMC has higher accuracy and good generalization performance.

INDEX TERMS Stereo matching cost, two-branch convolutional sparse representation, alternating direction method of multipliers, sparse representation.

I. INTRODUCTION

Stereo matching, also known as disparity mapping, is one of the key techniques in stereo vision research area. The core idea is to find all corresponding pixels in a stereo image pair. Stereo matching cost plays an important role in establishing visual matching relationship. Usually, the accuracy of the stereo matching method depends on the accuracy of the stereo matching cost. Commonly used stereo matching costs can be divided into two large categories, including pixel-wise and window-based matching costs. Pixel-wise matching costs include the absolute difference (AD) and truncated absolute difference (TAD) [1]. Window-based matching costs include follows: sum of squared difference (SSD) [2], [3], sum of absolute difference (SAD) [4], normalized cross correlation (NCC), zero mean normalized cross correlation (ZNCC) [5], census (Cen) [6], [7], etc.

To get better matching results, combinations or variations of the above window-based methods are proposed in the literature, such as combination of census and gradient based measures (Cen+G) [8], combination of sum of absolute difference and gradient-based measures (SAD+G) [9], and combination of absolute differences and census measures (AD+Cen) [10]. These stereo matching costs are used by some state-of-the-art stereo matching methods, and have been demonstrated to have very good performance in image regions with smooth terrain. However, they cannot handle regions that are lack of information, such as poorly texture regions, exposure variations, occlusion, depth discontinuities, etc.

Recently, stereo matching methods based on deep learning [11]–[19] had made significant pro-gress in the disparity estimation of stereo images, in which the most prominent and effective deep learning method is the deep convolutional neural network (CNN). Due to the powerful representation capability of deep CNN in poorly texture regions and repetitive texture regions, it has been employed to improve the accuracy of stereo matching. Žbontar and LeCun [18] first introduced CNN to measure the similarity between two image patches, which used the matching probability between two image patches as the stereo matching cost. Subsequently, a large number of stereo matching methods based on CNN were

The associate editor coordinating the review of this manuscript and approving it for publication was Jingchang Huang.

proposed. These CNN methods achieved better performance than conventional methods on challenging public benchmark data sets (such as KITTI [20] and Middlebury 2014 [2]). Authors of these methods suggested that it was unreliable to consider only the difference of photometry in pixels or hand-crafted image features for stereo matching cost. In contrast, CNN can learn more robust and discriminative features from images so that it can produce an improved stereo matching cost. Most CNN methods [11], [12], [14], [15], [18], [21]–[23] take stereo matching as a supervised learning task and are exploited to learn the stereo matching cost of two image patches. Only a small number of CNN methods [17], [24], [25] take stereo matching costs as unsupervised learning task, but their accuracy is significantly lower than that of the supervised CNN methods.

As pointed out in the literature [25]–[30], the supervised CNN based stereo matching costs require a large number of labeled training samples in the training process. There are following limitations on those methods. Firstly, the supervised CNN method [30] relies on ground truth disparity maps, however, the availability and creation of the ground truth disparity maps are not an easy job. In practice, although some dedicated ranging sensors, such as LIDAR and structured light sensors [31], can be used to generate the ground truth disparity maps, the former is relatively large and expensive, and ground truth disparity maps captured by LIDAR are generally sparse; the later does not work well under strong light environment. Secondly, existing CNN based stereo matching methods are usually trained and tested on the commonly used data sets, such as KITTI and Middlebury 2014, which are of specific scenes. For example, KITTI data set composes of many autonomous driving images with street views, while Middlebury 2014 data set provides image pairs with specific arranged environment or structured light. In general, there is few known public data set of general scenes with ground truth disparity maps for benchmark purpose, let alone real-life images. The main reasons are the high cost of the ground truth disparity maps acquisition (which requires a lot of manpower and expensive instrument resources) and the prone to pixel-level annotation errors of manually created ground truth disparity maps. The performance of the CNN based stereo matching costs is restricted by the amount of training data with accurate ground truth disparity maps. Those methods trained and tested on the commonly used data sets, such as KITTI and Middlebury 2014, perform poorly in general scenes or real-life images. Therefore, it is necessary to develop a new stereo matching cost that is minimally dependent on the ground truth disparity maps.

In recent years, sparse representation (SR) [32]–[35] and convolutional sparse representation (CSR) [36]–[39] have attracted widespread interests in the field of visual recognition, and have made great success. SR can construct sparse representation coefficients from the main or essential features of the represented object, and these sparse representation coefficients have good discriminability. CSR can learn the convolutional filter bank from stereo image pairs in an unsupervised manner, but it requires a large number of convolution kernels to characterize the geometric features of stereo image pairs when performing convolution sparse decomposition of stereo image pairs. It can easily lead to the redundancy of the convolution kernels, which increases the computational complexity.

Inspired by two-branch CNN, this paper first proposes a new two-branch convolutional sparse representation (TCSR) model. This model imposes a ℓ_2-norm constraint on the extracted features, so that the same feature is prevented from being represented by different convolution kernels, which reduces the redundancy of the convolution kernels, thereby reducing the computational complexity. Then, based on the TCSR model, an unsupervised stereo matching cost is devised, and it does not rely on the ground truth disparity maps, and its matching accuracy is higher than that of conventional stereo matching costs.

The main contributions and novelties of this paper are as follows.

- A new TCSR model is proposed to learn the convolutional filter bank from the left and right images. This model imposes a ℓ_2-norm constraint on the extracted features, so that the extracted features of the left and right images are as close as possible, thereby further constraining the global convolution kernels. In this way, the same feature is prevented from being represented by different convolution kernels, which reduces the redundancy of the convolution kernels and the computational complexity, and achieves higher matching accuracy.
- A simple and efficient iterative algorithm is devised for the TCSR model. Due to the proposed TCSR model belongs to the tri-convex (joint non-convex) optimization problem (conventional CSR models belong to bi-convex optimization problem), which is usually difficult to solve. The iterative algorithm we devised can solve this tri-convex optimization problem. The convergence is also discussed theoretically to ensure the effectiveness of the proposed algorithm.
- Based on TCSR model, a new unsupervised stereo matching cost is proposed. It can not only preserve the geometric details of stereo images and produce smooth disparity maps, but also achieve very good performance on challenging regions, such as exposure changes and occlusion regions. Providing any binocular scenes without ground truth disparity maps, the disparity maps can be generated automatically.

The rest of this paper is organized as follows. Section II presents the TCSR model, iterative algorithm and corresponding convergence analysis. In addition, this section also introduces the unsupervised stereo matching cost based on the TCSR model. Afterward, experimental results are shown in Section III. Section IV concludes and discusses the whole work.

II. TCSR FOR STEREO MATCHING

In this section, we build a two-branch convolutional sparse representation (TCSR) model, and design a iterative algorithm to solve this model. Then, the TCSR is applied to the

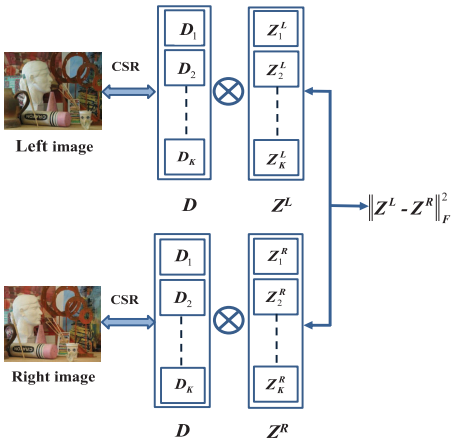

FIGURE 1. The overview of the TCSR model.

stereo matching, and the unsupervised stereo matching cost is constructed.

A. TCSR MODEL

Given a training set including N rectified stereo image pairs $\{\boldsymbol{I}_j^L, \boldsymbol{I}_j^R\}_{j=1}^N$, TCSR can learn the shared convolution kernels $\{\boldsymbol{D}_k^T\}_{k=1}^K$ from stereo image pairs, Fig.1 provides the overview of the TCSR model. We view $\boldsymbol{D} = [\boldsymbol{D}_1^T, \ldots, \boldsymbol{D}_K^T]^T$ as convolutional filter bank, where $\{\boldsymbol{D}_k^T\}_{k=1}^K$ is a set including K convolution kernels. In the training phase, convolutional filter bank $\boldsymbol{D}$ is learned from training set $\{\boldsymbol{I}_j^L, \boldsymbol{I}_j^R\}_{j=1}^N$ by minimizing the following objective function:

$$\begin{aligned}
&\min_{\boldsymbol{Z}_j^L, \boldsymbol{Z}_j^R, \boldsymbol{D}} \{\mathcal{C}(\boldsymbol{Z}_j^L, \boldsymbol{Z}_j^R, \boldsymbol{D}) \\
&= \sum_{j=1}^{N} (\underbrace{\frac{1}{2}\|\boldsymbol{I}_j^L - \sum_{k=1}^{K} \boldsymbol{D}_k \otimes \boldsymbol{Z}_{jk}^L\|_2^2}_{\textbf{The first term}} \\
&+ \underbrace{\lambda \sum_{k=1}^{K} \|\boldsymbol{Z}_{jk}^L\|_1}_{\textbf{The second term}} + \underbrace{\frac{1}{2}\|\boldsymbol{I}_j^R - \sum_{k=1}^{K} \boldsymbol{D}_k \otimes \boldsymbol{Z}_{jk}^R\|_2^2}_{\textbf{The third term}} \\
&+ \underbrace{\lambda \sum_{k=1}^{K} \|\boldsymbol{Z}_{jk}^R\|_1}_{\textbf{The fourth term}} + \underbrace{\frac{\beta}{2} \sum_{k=1}^{K} \|\boldsymbol{Z}_{jk}^L - \boldsymbol{Z}_{jk}^R\|_2^2)}_{\textbf{The fifth term}}\} \\
&s.t.\ \|\boldsymbol{D}_k\|_2^2 \le 1,\ \forall k = 1, 2, \ldots, K, \qquad (1)
\end{aligned}$$

where $\otimes$ denotes two-dimensional discrete convolution operator. The variables $\boldsymbol{I}_j^L$ (or $\boldsymbol{I}_j^R$) $\in \mathcal{R}^{M\times 1}$ and $\boldsymbol{Z}_{jk}^L$ (or $\boldsymbol{Z}_{jk}^R$) $\in \mathcal{R}^{M\times 1}$ are vectorized image and feature map, respectively. $\boldsymbol{D}_k \in \mathcal{R}^{S\times 1}$ represents the vectorized convolution kernel. λ and β are regularization parameters. The first and the third term in (1) represent the reconstruction errors. The second and the fourth term make feature maps sparse enough. The fifth term can control the feature map $\boldsymbol{Z}_{jk}^L$ extracted from the left image $\boldsymbol{I}_j^L$ to approximate the feature map $\boldsymbol{Z}_{jk}^R$ extracted from the right image $\boldsymbol{I}_j^R$ under the same convolution kernel, so that the same feature is prevented from being represented by different convolution kernels, which reduces the redundancy of the convolution kernels. Note that here $\|\boldsymbol{Z}\|_1$ represents the entry-wise vector ℓ_1 norm.

B. ITERATIVE ALGORITHM

For optimization problem (1), any two variables in (1) are fixed, then (1) is convex with respect to the remaining variable, so that the proposed objective function is tri-convex optimization (joint non-convex) problem. Using the fixed point strategy to solve $\boldsymbol{Z}_j^L$, $\boldsymbol{Z}_j^R$ and $\boldsymbol{D}$ in (1), it can divide (1) into the following three sub-problems:

1) $\boldsymbol{Z}_j^L \leftarrow arg\min_{\boldsymbol{Z}_j^L} \mathcal{C}(\boldsymbol{Z}_j^L, \boldsymbol{Z}_j^R, \boldsymbol{D})$
2) $\boldsymbol{Z}_j^R \leftarrow arg\min_{\boldsymbol{Z}_j^R} \mathcal{C}(\boldsymbol{Z}_j^L, \boldsymbol{Z}_j^R, \boldsymbol{D})$
3) $\boldsymbol{D} \leftarrow arg\min_{\boldsymbol{D}} \mathcal{C}(\boldsymbol{Z}_j^L, \boldsymbol{Z}_j^R, \boldsymbol{D})$

Due to each sub-problem is convex, so we can use alternating direction method of multipliers (ADMM) to solve it. The specific solution process of the three sub-problems will be introduced in supplementary materials. Only the update formulas of the three variables $\boldsymbol{Z}_j^L$, $\boldsymbol{Z}_j^R$ and $\boldsymbol{D}$ are shown here.

• **Updating $\boldsymbol{Z}_j^L$:** in order to improve computational efficiency, we can update $\boldsymbol{Z}_j^L$ at the t-th step in the Fourier domain:

$$(\hat{\boldsymbol{Z}}_j^L)_t \leftarrow (\bar{\boldsymbol{D}}^H\bar{\boldsymbol{D}} + (\rho+\beta)\boldsymbol{E}_{KM})^{-1}(\bar{\boldsymbol{D}}^H\hat{\boldsymbol{I}}_j^L + \rho\hat{\boldsymbol{X}}_j^L - \hat{\boldsymbol{U}}_j^L + \beta\hat{\boldsymbol{Z}}_j^R), \qquad (2)$$

where $\boldsymbol{E}_{KM} \in \mathcal{R}^{KM\times KM}$ is the identity matrix. Then $\boldsymbol{Z}_j^L$ can be reconstructed back from $\hat{\boldsymbol{Z}}_j^L$ by taking the inverse Fourier transform, namely, $\boldsymbol{Z}_j^L = \mathcal{F}^{-1}(\hat{\boldsymbol{Z}}_j^L)$, in which $\mathcal{F}^{-1}$ is the inverse Fourier transform.

• **Updating $\boldsymbol{Z}_j^R$:** solving $\boldsymbol{Z}_j^R$. The solution process of $\boldsymbol{Z}_j^R$ is the same as that of $\boldsymbol{Z}_j^L$, which can update by

$$(\hat{\boldsymbol{Z}}_j^R)_t \leftarrow (\bar{\boldsymbol{D}}^H\bar{\boldsymbol{D}} + (\gamma+\beta)\boldsymbol{E}_{KM})^{-1}(\bar{\boldsymbol{D}}^H\hat{\boldsymbol{I}}_j^R + \gamma\hat{\boldsymbol{X}}_j^R - \hat{\boldsymbol{U}}_j^R + \beta\hat{\boldsymbol{Z}}_j^L), \qquad (3)$$

$\boldsymbol{Z}_j^R$ can be reconstructed back by taking the inverse Fourier transform of $\hat{\boldsymbol{Z}}_j^R$.

• **Updating $\boldsymbol{D}$:** in order to improve computational efficiency, we also update $\boldsymbol{D}$ in the Fourier domain:

$$\hat{\boldsymbol{D}}_t \leftarrow ((\sum_{j=1}^{N} \hat{\boldsymbol{Z}}_j^{*H}\hat{\boldsymbol{Z}}_j^*) + \eta\boldsymbol{E}_{KM})^{-1}((\sum_{j=1}^{N} \hat{\boldsymbol{Z}}_j^{*H}\hat{\boldsymbol{I}}_j) + \eta\hat{\boldsymbol{G}} - \hat{\boldsymbol{V}}), \qquad (4)$$

then $\boldsymbol{D} = \mathcal{F}^{-1}(\hat{\boldsymbol{D}})$.

To sum up, the solution procedure for the TCSR model in (1) is outlined in **Algorithm 1**.

Algorithm 1 TCSR Model

Input: Training set $\boldsymbol{I} = \{\boldsymbol{I}_j^L, \boldsymbol{I}_j^R\}_{j=1}^N$.

Output: convolutional filter bank $\boldsymbol{D} = \mathcal{F}^{-1}(\hat{\boldsymbol{D}})$.

Initialize: $\boldsymbol{D}_0, \boldsymbol{U}_j^L, \boldsymbol{U}_j^R \sim \mathcal{N}(0,1)$, $\boldsymbol{Z}_j^L = 0$, $\gamma = \beta = 1$, $\rho = 1$, $\eta = 1$, $\boldsymbol{X}_j^L = 0$; $\boldsymbol{Z}_j^R = 0$, $\boldsymbol{X}_j^R = 0$;
Precompute Fourier transforms $\hat{\boldsymbol{I}} = \mathcal{F}(\boldsymbol{I})$, $\hat{\boldsymbol{D}}_0 = \mathcal{F}(\boldsymbol{D}_0)$, $\hat{\boldsymbol{U}}_j^L = \mathcal{F}(\boldsymbol{U}_j^L)$, $\hat{\boldsymbol{X}}_j^L = \mathcal{F}(\boldsymbol{X}_j^L)$, $\hat{\boldsymbol{G}}_0 = \hat{\boldsymbol{D}}_0$, $\hat{\boldsymbol{V}}_0 = \hat{\boldsymbol{G}}_0$, $\hat{\boldsymbol{Z}}_j^L = \mathcal{F}(\boldsymbol{Z}_j^L)$; $\hat{\boldsymbol{U}}_j^R = \mathcal{F}(\boldsymbol{U}_j^R)$, $\hat{\boldsymbol{X}}_j^R = \mathcal{F}(\boldsymbol{X}_j^R)$, $\hat{\boldsymbol{Z}}_j^R = \mathcal{F}(\boldsymbol{Z}_j^R)$;
repeat
for $j = 1, 2, \ldots, N$
    Update $\hat{\boldsymbol{Z}}_j^L$ by (2), compute $\boldsymbol{Z}_j^L = \mathcal{F}^{-1}(\hat{\boldsymbol{Z}}_j^L)$;
    Solve $\boldsymbol{X}_j^L$ by the soft thresholding operator;
    Update $\boldsymbol{U}_j^L := \boldsymbol{U}_j^L + \rho(\boldsymbol{Z}_j^L - \boldsymbol{X}_j^L)$;
    Update $\hat{\boldsymbol{Z}}_j^R$ by (3), compute $\boldsymbol{Z}_j^R = \mathcal{F}^{-1}(\hat{\boldsymbol{Z}}_j^R)$;
    Update $\boldsymbol{X}_j^R$, $\boldsymbol{U}_j^R$;
end for
Update $\hat{\boldsymbol{D}}$ by (4), compute $\boldsymbol{D} = \mathcal{F}^{-1}(\hat{\boldsymbol{D}})$;
Update $\boldsymbol{G}$, $\boldsymbol{V}$;
until convergence (maximum iterations T reached or objective function $\leq$ threshold)

C. CONVERGENCE ANALYSIS

The proposed objective function in (1) belongs to the non-convex optimization problem, which is usually difficult to solve. In the previous section, we devise an iterative algorithm to solve (1). Next, it is proved by Theorem 1 that this iterative algorithm is locally convergent, the rigorous mathematical proof is given in the supplementary materials.

Theorem 1: The sequence $\{\boldsymbol{W}_t, \boldsymbol{U}_t^L, \boldsymbol{U}_t^R, \boldsymbol{V}_t\}$ generated by **Algorithm 1** has at least one accumulation point $\{\boldsymbol{W}^*, (\boldsymbol{U}^L)^*, (\boldsymbol{U}^R)^*, \boldsymbol{V}^*\}$. Then $\boldsymbol{W}^*$ is a stationary point of (1) under the condition that the Lagrangian multiplier sequence $\{\boldsymbol{U}_t^L, \boldsymbol{U}_t^R, \boldsymbol{V}_t\}$ is bounded and satisfies

$$\sum_{t=0}^{\infty}(\|\boldsymbol{U}_{t+1}^L - \boldsymbol{U}_t^L\|_2^2 + \|\boldsymbol{U}_{t+1}^R - \boldsymbol{U}_t^R\|_2^2 + \|\boldsymbol{V}_{t+1} - \boldsymbol{V}_t\|_2^2) < \infty. \tag{5}$$

D. COMPLEXITY ANALYSIS

The computational complexity of the algorithm is discussed in this subsection. We can obtain the total computational complexity of TCSR for each pair image by analyzing the computational complexity of each step in (1). In fact, the most expensive part in (1) is updating $\boldsymbol{Z}_j^L, \boldsymbol{Z}_j^R$ and $\boldsymbol{D}$ in one iteration, whose computational complexity can be analyzed as follows.

(I) Updating $\boldsymbol{Z}_j^L$. The computational complexity of updating $\boldsymbol{Z}_j^L$ is

$$\underbrace{\mathcal{O}(KM\log_2 M)}_{\textbf{Fast Fourier Transforms}} + \underbrace{\mathcal{O}(MK^3)}_{\textbf{Linear Systems}}, \tag{6}$$

where $\mathcal{O}(KM\log_2 M)$ is the computational complexity of fast Fourier transforms (FFTs), and the computational complexity of the linear systems is $\mathcal{O}(MK^3)$.

(II) Updating $\boldsymbol{Z}_j^R$. The computational complexity of updating $\boldsymbol{Z}_j^R$ is the same as updating $\boldsymbol{Z}_j^L$, which is $\mathcal{O}(KM\log_2 M) + \mathcal{O}(MK^3)$.

(III) Updating $\boldsymbol{D}$. The computational complexity of updating $\boldsymbol{D}$ is

$$\underbrace{\mathcal{O}(KM\log_2 M)}_{\textbf{Inverse Fourier Transforms}} + \underbrace{\mathcal{O}(MK^3)}_{\textbf{Linear Systems}}, \tag{7}$$

where computational complexity of inverse Fourier transforms is $\mathcal{O}(KM\log_2 M)$.

Therefore, we can conclude that the total computational complexity of TCSR for each pair image is estimated to be $\mathcal{O}(3KM\log_2 M) + \mathcal{O}(3MK^3)$ by summing (I), (II) and (III).

Remark 1: Given N image pairs $\{\boldsymbol{I}_j^L, \boldsymbol{I}_j^R\}_{j=1}^N$, we can conclude that the total computational complexity of TCSR is $\mathcal{O}(3NKM\log_2 M) + \mathcal{O}(3NMK^3)$, and the computational complexity of each branch in TCSR is $\mathcal{O}(2NKM\log_2 M) + \mathcal{O}(2NMK^3)$ in the worst case. Compared with the computational complexity of conventional CSR models $\mathcal{O}(4NKM\log_2 M) + \mathcal{O}(4NMK^3)$, the total computational complexity of TCSR and the computational complexity of each branch in TCSR are lower. In terms of the computational complexity of the two methods, the number K of convolution kernels directly affects the computational complexity, so the redundancy of convolution kernels will increase the computational complexity. From this perspective, our algorithm has more advantages due to less redundancy.

E. STEREO MATCHING

1) UNSUPERVISED STEREO MATCHING COST

The TCSR is applied to the stereo matching, and the unsupervised stereo matching cost is constructed. Given the image patch $\boldsymbol{P}^L(q)$ is indicated a patch from the left image $\boldsymbol{Y}^L$, centered at $q = (x, y)$, and $\boldsymbol{P}^R(q-d)$ is denoted as a patch from the right image $\boldsymbol{Y}^R$, centered at $q-d = (x-d, y)$, we can construct the unsupervised stereo matching cost (USMC) as follows:

$$C(q,d) = \|\boldsymbol{w}^L(\mathbf{q}) - \boldsymbol{w}^R(q-d)\|_1. \tag{8}$$

Here, $\boldsymbol{w}^L(\mathbf{q})$ and $\boldsymbol{w}^R(q-d)$ can be calculated by the following formula:

$$arg\min_{\boldsymbol{w}} \frac{1}{2}\|\bar{\boldsymbol{P}} - \tilde{\boldsymbol{D}}\boldsymbol{w}\|_2^2 + \alpha\|\boldsymbol{w}\|_1, \tag{9}$$

where $\bar{\boldsymbol{P}} \in \{\bar{\boldsymbol{P}}^L(q), \bar{\boldsymbol{P}}^R(q-d)\}$, $\boldsymbol{w} \in \{\boldsymbol{w}^L(q), \boldsymbol{w}^R(q-d)\}$, $\boldsymbol{w}^L(q)$, $\boldsymbol{w}^R(q-d) \in \mathcal{R}^{K\times 1}$. The image patches $\boldsymbol{P}^L(q)$ and $\boldsymbol{P}^R(q-d)$ are expanded into column vectors, and denoted as $\bar{\boldsymbol{P}}^L(q) \in \mathcal{R}^{S\times 1}$ and $\bar{\boldsymbol{P}}^R(q-d) \in \mathcal{R}^{S\times 1}$, respectively. Since $\boldsymbol{D}$ is the vectorized convolutional filter bank obtained by the TCSR model, we need to rewrite $\boldsymbol{D}$ so that each of its columns is a filter, i.e., $\tilde{\boldsymbol{D}} \in \mathcal{R}^{S\times K}$. We can solve sparse representation coefficients in (9) by using least angle regression.

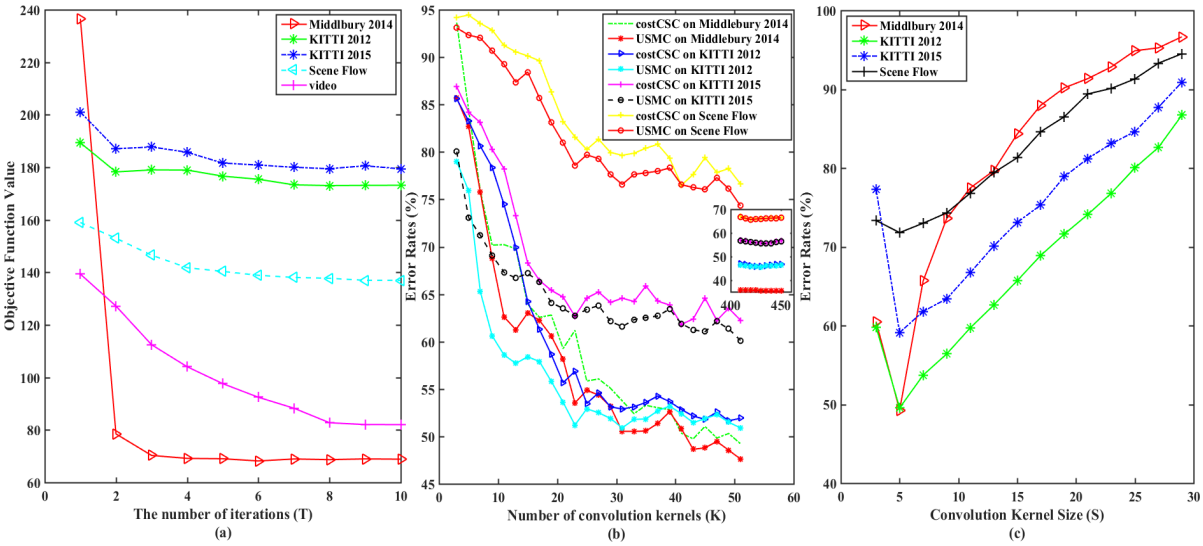

FIGURE 2. Parameters analysis. (a) The objective function value varies with different T on different data sets. (b) Error rates of the USMC and the costCSC vary along with the number of convolution kernels K on different data sets. The sub-figure shows that error rates of the USMC and the costCSC vary along with the number of convolution kernels $K \in \{410, 415, \ldots, 450\}$. (c) Error rates of the USMC vary along with the convolution kernel size S on different data sets.

III. EXPERIMENTS

A. DATASETS DESCRIPTION

The Middlebury 2014 data set [2], [40] consists of 15 training images and 15 test images without ground truth disparity maps. These images are acquired by different stereo systems and contain different artificial indoor scenes. They are available in three resolutions, namely, full (F), half (H), and quarter (Q), here we use half-resolution images to do the experiments.

KITTI data set [20], [41] contains two sub-datasets (i.e. KITTI 2012 and KITTI 2015), where KITTI 2012 data set contains 194 training images and 195 test images without ground truth disparity maps; KITTI 2015 data set contains 200 training images and 200 test images without ground truth disparity maps. These images are captured from real world dataset with street views.

The Scene Flow data set [42] contains more than 39000 stereo frames in 960 $\times$ 540 pixel resolution, rendered from various synthetic sequences. It has three sub-datasets, i.e., FlyingThings3D, Driving, and Monkaa. The monocular video clip is taken from real road navigation, which contains 141 images without ground truth disparity map.

B. PARAMETER ANALYSIS

Before doing the experiment, we need to initialize the model parameters. The parameters involved in this paper mainly include: the maximum iteration T, the convolution kernel size S, and number of convolution kernels K.

For convolutional filter bank learning, we attempt to keep the same settings as in the paper [37] except for the maximum iteration T, the convolution kernel size S and number of convolution kernels K. In order to select initial parameter T, we conduct experiments using different data sets. The value of T is selected within the range of $\{1, 2, \ldots, 10\}$. The objective function value in (1) with different values of T are shown in Fig. 2(a).

In Fig. 2(a), it can be seen that the objective function value in (1) first decreases with increasing of T and then keeps unchanged. The optimal iterations' number of the proposed method on five data sets is almost 8. It suggests the good convergence performance of the proposed algorithm. Taking into account the computational cost of the TCSR model, a moderate value of T can be chosen in practice.

For the convolution kernel size S, it directly affects the quality of the extracted features. In order to analyze the role of parameter S in the performance of the stereo matching cost on different data sets, we traverse the range of the patch size from $\{3, 5, \ldots, 29\}$. The error rates of the USMC with different values of S are shown in Fig. 2(c).

In Fig. 2(c), we can see that the error rates of the USMC first decrease with increasing of S and then increase with increasing of S. This is because small value of S means the image information cannot be sufficiently expressed, resulting in poor extracted features. When using large convolution kernel, information expression becomes redundant, resulting in an increase in error rate. Moreover, the size of the convolution kernel is zero-padded to M before applying the fast Fourier transform (FFT). From Remark 1, we know that a small S means less computational cost. In order to control the amount of calculation and obtain a higher matching accuracy, a moderate value of S can be selected in practice.

As for K, the error rates of USMC and the error rates of costCSC with different values of K on different data sets are shown in Fig. 2(b), where costCSC in Fig. 2(b) represents the stereo matching cost based on conventional convolutional sparse representation models. It can be seen that they are

fluctuating, but error rates have a tendency to decrease all the time. It can also be seen in the sub-figure (Fig. 2(b)) that error rates of the USMC and the costCSC on different data sets are minimized and remain almost unchanged when K ($K \in \{410, 415, \ldots, 450\}$) is large. Finally, $K = 440$ is selected in terms of matching accuracy and computational efficiency.

Based on the above parameter analysis, we initialize the parameters as follows: $T = 8$, $S = 5$ and $K = 440$.

C. ABLATION STUDIES

1) ABLATION STUDIES FOR TSCR MODEL

We use the following notations: costCSC represents the stereo matching cost based on conventional convolutional sparse representation models; the TCFB is for the convolutional filter bank learned by the TCSR model; CCFB represents the convolutional filter bank learned by the conventional convolutional sparse representation models.

In order to verify the effectiveness of the proposed TCSR, we compare TCSR and CSR from two aspects: error rate and the convolution kernels.

Comparison of error rate of TCSR and CSR: the TCSR model can effectively characterize the geometric features of stereo images with fewer convolution kernels K, while conventional CSR models require more convolution kernels K. In order to analyze the role of parameter K in the precision of stereo matching cost and verify the effectiveness of the TCSR model, we conduct the first set of comparison experiment using different data sets.

The error rates of USMC and costCSC with different values of K are shown in Table 1. Overall, the error rates of USMC is lower than those of costCSC under the same number of convolution kernels. It shows that the convolutional filter bank learned by the TCSR model has stronger representation ability than that learned by the conventional CSR models. Note that no cost aggregation and disparity refinement step are applied in this experiment, because we hope that the original results will provide a more direct assessment for the different stereo matching costs.

From Table 1, it can be observed that the error rates of the stereo matching cost are directly affected by the number of convolution kernels and the error rates of USMC are lower than that of costCSC. Moreover, when $K = 13$, the error rates of USMC is even lower than the error rates of costCSC when $K = 19$. It shows that the error rates of the USMC with fewer convolution kernels are lower than that of the costCSC with more convolution kernels. In Table 1, it can also be observed that the proposed TCSR takes less time than the conventional CSR, and the conventional CSR takes about 2.5 times as much time as the TCSR.

Comparison of convolution kernels trained by TCSR and CSR: for visual comparison, Fig. 3 shows TCFB and CCFB with the same number of convolution kernels, respectively. To find the difference between TCFB and CCFB, the following Bhattacharyya distance formula is used to measure the similarity between convolution kernel $\boldsymbol{D}_i$ and $\boldsymbol{D}_j$ ($i, j \in \{1, 2, \ldots, K\}$) in TCFB or CCFB.

TABLE 1. The comparisons of the number of convolution kernels and error rates, the number of convolution kernels and GPU runtime (s) between USMC and costCSC performed on different data sets. The best results are shown in bold.

Datasets	Methods	The number of convolution kernels & error rates (%)					
		3	5	9	13	17	19
Middlebury 2014	costCSC	93.7	84.5	70.2	69.8	62.6	62.8
	Ours	**85.7**	**82.7**	**68.8**	**61.3**	**62.2**	**60.6**
KITTI 2012	costCSC	85.6	83.3	78.4	70.0	61.3	58.6
	Ours	**79.0**	**75.9**	**60.6**	**57.7**	**57.9**	**55.8**
KITTI 2015	costCSC	86.9	84.2	80.3	73.3	66.5	65.4
	Ours	**80.1**	**73.1**	**69.1**	**65.3**	**65.2**	**64.1**
Scene Flow	costCSC	94.2	94.4	92.9	90.6	89.6	86.3
	Ours	**93.1**	**92.3**	**90.7**	**86.1**	**85.7**	**83.1**
Datasets	Methods	The number of convolution kernels & GPU runtime (s)					
		3	5	9	13	17	19
Middlebury 2014	costCSC	2.8	3.2	3.5	3.8	4.3	5.3
	Ours	**1.1**	**1.2**	**1.4**	**1.5**	**1.7**	**2.1**
KITTI 2012	costCSC	5.9	6.1	6.4	6.9	6.6	7.2
	Ours	**2.3**	**2.4**	**2.5**	**2.7**	**2.6**	**2.8**
KITTI 2015	costCSC	6.4	6.6	6.4	6.9	6.7	7.3
	Ours	**2.5**	**2.6**	**2.5**	**2.7**	**2.7**	**2.8**
Scene Flow	costCSC	5.1	5.3	5.5	5.6	5.9	6.1
	Ours	**2.1**	**2.1**	**2.2**	**2.2**	**2.3**	**2.4**

$$\boldsymbol{similarity}(\boldsymbol{D}_i, \boldsymbol{D}_j) = 1 - \sqrt{1 - \frac{\sum_g \sqrt{H_{\boldsymbol{D}_i}(g) H_{\boldsymbol{D}_j}(g)}}{\sqrt{\sum_g H_{\boldsymbol{D}_i}(g) \sum_g H_{\boldsymbol{D}_j}(g)}}}, \tag{10}$$

where $H_{\boldsymbol{D}_i}(g)$ and $H_{\boldsymbol{D}_j}(g)$ represent the number of pixels with gray value g in convolution kernel $\boldsymbol{D}_i$ and $\boldsymbol{D}_j$, respectively. Equation (10) can be used to calculate the similarity measurement matrices of TCFB and CCFB, as shown in the matrix on the right of Fig. 3(a) and Fig. 3(b).

Quantitatively, the similarity between the convolution kernels in CCFB is greater than that between the convolution kernels in TCFB. This shows that the TCSR model can decrease the redundancy of convolution kernel, thus reduce the computational complexity. Due to the limitation of the length of the paper, we only analyzed the properties of convolution kernels learned by the two CSR models on Middlebury 2014 in this paper, and the same conclusion can be drawn on other data sets.

2) ABLATION STUDIES FOR THE PROPOSED STEREO MATCHING COST

In this section, the USMC is compared with popular window-based stereo matching costs (i.e., Cen, SAD+G, SSD, NCC, AD+Cen, ZNCC and Cen+G) on different data sets. Among these selected methods, AD+Cen [10], Cen+G [8] and SAD+G [9] are employed by some state-of-the-art stereo matching methods on the Middlebury 2014 dataset [2], and Cen+G is used by one of the state-of-the-art stereo matching methods on the outdoor KITTI data sets. In addition, in order to analyze the effect of filter banks obtained

TABLE 2. Ablation studies for the different stereo matching costs with or without cost aggregation and post-processing.

	Cost Aggregation	Post-processing	Cen	SAD+G	SSD	NCC	AD+Cen	ZNCC	Cen+G	Ours
Middlebury 2014	Yes	No	51.3	65.8	68.0	64.7	51.4	44.3	53.2	**32.4**
	Yes	Yes	35.4	48.5	39.1	35.4	35.2	28.7	36.5	**19.1**
KITTI 2012	Yes	No	47.7	60.4	56.3	52.4	39.4	48.3	51.7	**17.5**
	Yes	Yes	17.2	31.7	31.5	26.3	15.3	18.7	22.8	**6.4**
KITTI 2015	Yes	No	43.9	63.6	51.6	44.4	30.8	46.2	43.6	**16.6**
	Yes	Yes	16.3	38.5	28.4	26.1	15.4	17.5	15.3	**6.2**
FlyingThings3D	Yes	No	58.3	60.2	62.2	61.6	63.2	48.9	45.4	**57.0**
	Yes	Yes	27.6	29.5	38.6	35.3	36.3	32.9	28.8	**24.8**
Driving	Yes	No	60.2	63.7	64.7	59.4	54.9	53.1	57.2	**46.8**
	Yes	Yes	41.3	45.9	47.1	38.7	36.2	34.2	38.5	**17.4**
Monkaa	Yes	No	65.4	61.6	68.2	65.5	58.4	62.6	55.6	**53.3**
	Yes	Yes	45.1	40.3	50.9	47.2	38.4	43.8	34.7	**21.7**

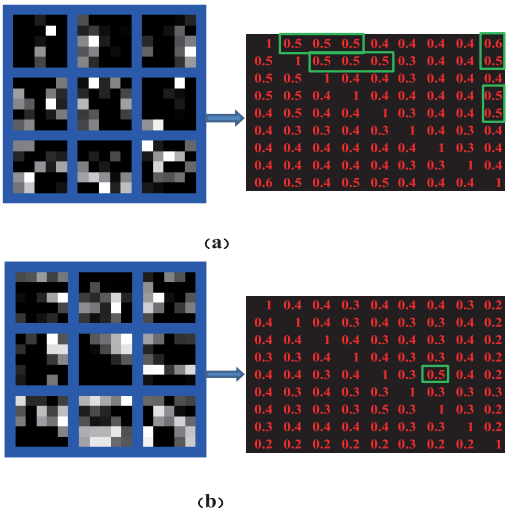

FIGURE 3. TCFB and CCFB are learned by the TCSR model and conventional CSR models on training data set from the Middlebury 2014, respectively. (a) **Left:** CCFB is learned by the conventional CSR models. **Right:** the similarity measurement matrix of CCFB. (b) **Left:** TCFB is learned by the TCSR model. **Right:** the similarity measurement matrix of TCFB. Note that the larger values in the similarity measurement matrix are marked with green rectangles.

by different training methods on the error of the proposed USMC, we conducted two sets of experiments: (1) for each data set of Middlebury 2014, KITTI and Scene Flow, the filter banks are trained, respectively, and the experimental results are shown in the penultimate column (Ours (each)) in Table 3. (2) The data sets of Middlebury 2014, KITTI and Scene Flow are combined into one dataset to train a single filter bank, and the experimental results are shown in the last column (Ours) in Table 3. From the results of the last two columns in Table 3, the error rate of the USMC by training a single filter bank on the Middlebury 2014, KITTI, and Scene Flow datasets is lower than that of the USMC by training the filter bank on each dataset separately. Therefore, subsequent experiments in the following sections will use the experimental method of group (2) above.

For a fair comparison, the window size is chosen to be 5×5. We compare error rates of different window-based stereo matching costs on Middlebury 2014, KITTI 2012, KITTI 2015 and Scene Flow data set (including FlyingThings3D, Driving and Monkaa). The quantitative results are shown in Table 3.

TABLE 3. Quantitative evaluation (average error rates) of the different window based stereo matching costs on different data sets.

Datasets	Cen	SAD+G	SSD	NCC	AD+Cen	ZNCC	Cen+G	Ours (each)	Ours
Middlebury 2014	71.0	85.0	74.6	84.2	71.2	64.1	73.7	53.7	**53.4**
KITTI 2012	59.0	82.0	65.8	61.0	58.7	67.6	71.0	56.2	**54.9**
KITTI 2015	55.7	78.4	63.0	62.6	52.8	68.5	66.1	45.2	**44.2**
FlyingThings3D	79.7	80.3	74.7	79.7	82.6	69.5	67.3	63.6	**62.7**
Driving	73.4	75.0	76.2	77.5	65.4	64.1	70.7	52.2	**51.2**
Monkaa	79.2	75.3	80.6	84.2	69.1	72.6	60.1	55.4	**55.0**

It can be seen that average error rate (see the last two column of Table 3) of our method is significantly lower than that of conventional window-based stereo matching costs on different data sets. Note that all the results in Table 3 are not processed by the cost aggregation and post-processing, which can provide a more direct assessment of different stereo matching costs.

3) ABLATION STUDIES FOR COST AGGREGATION AND POST-PROCESSING

By comparing the error rate of the proposed USMC and the conventional stereo matching costs on different datasets, we analyze the impact of cost aggregation [43], [44] and post-processing [9], [45] on the performance of the stereo matching costs, and the results are shown in Table 2. As listed in Table 2, the USMC with cost aggregation and post-processing improves the disparity results, which shows that the USMC with cost aggregation and post-processing has achieved a lower error rate than that without cost aggregation and post-processing. In addition, we observe that utilizing the cost aggregation to encourage local smoothness further helps to improve the results. This is due to the fact that such techniques eliminate small isolated noisy areas. While the post-processing focuses on occlusions and sub-pixel enhancement, which adds extra robustness to non-textured areas by fitting slanted planes. Our method with combination of the cost aggregation and post-processing achieves a lower error rate.

Fig. 4 shows the qualitative comparison results on different datasets (seen first six columns). It can be seen that the

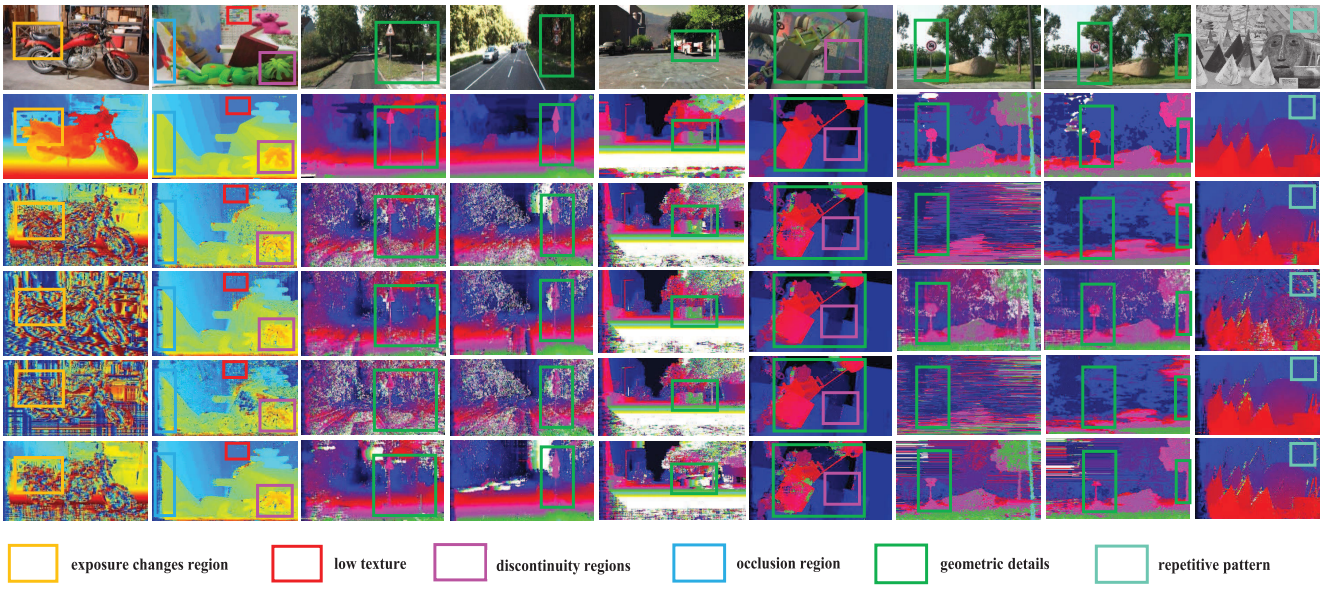

FIGURE 4. Qualitative comparison results on different datasets. From first to second column: input color images from Middlebury 2014; third to fourth column: input color images from KITTI; fifth to sixth column: input color images from Scene Flow; and seventh to eighth column: input color images from monocular video clip without ground truth disparity map; last column: input plain color image from Middlebury datasets. From second row to sixth row: the results achieved by our method, [2], [4], [8] and [6], respectively.

TABLE 4. Quantitative evaluation (error tolerance: 3 pixel) of different methods on KITTI 2012 test dataset and KITTI 2015 test dataset.

Supervised Methods	KITTI 2012		KITTI 2015	
	Out-noc	Avg-all	D1-bg	D1-all
MC-CNN-WS fst [15]	13.9%	-	14.1%	-
MC-CNN-fst [18]	15.4%	-	15.4%	-
Deep Embed [23]	8.9%	4.9 px	7.3%	2.0%
Content-CNN Unary [11]	6.6%	3.3 px	7.1%	1.9%
DispNet [42]	-	-	7.2%	1.8%
DSGCA [46]	10.7%	12.3%	-	-
RBE [47]	-	-	-	17.4%
Chen et al. [48]	-	-	-	4.3%
Unsupervised Methods	**KITTI 2012**		**KITTI 2015**	
	Out-noc	Avg-all	D1-bg	D1-all
USCNN [17]	-	-	11.2%	-
MBM [49]	7.9%	1.4 px	6.1%	7.4%
Yu et al. [50]	-	-	8.6%	9.9%
Zhou et al. [25]	-	-	6.8%	1.6%
OASM-Net [24]	**6.4%**	2.0 px	6.9%	9.0%
ELAS [51]	8.2%	1.7 px	7.9%	9.7%
CostFilter [9]	20.0%	5.4 px	17.5%	18.4%
Li et al. [52]	9.3%	1.8 px	-	-
MTS [53]	-	-	-	4.4 %
Ours	**6.4%**	**1.3** px	**6.2%**	**1.2%**

"-" **means that no experiments have been done on KITTI 2012 or KITTI 2015 dataset**.

proposed method produces smooth and dense disparity maps. Our method not only preserves geometry details near depth discontinuities, but also performs well on challenging regions such as exposure variations and occlusion regions.

D. EVALUATION ON THE REAL WORLD SCENES WITHOUT GROUND TRUTH DISPARITY MAPS

To test generalization ability of the proposed method, we further evaluate it on a monocular video clip without ground truth disparity map, and use the model trained on KITTI. The results are shown in Fig. 4 (seen last two columns). From Fig. 4, it can be seen that: the method of [2], [8] and [6] produce disparity maps with poor visual effects and lose geometric details; although the method of [4] can keep geometry details well, it cannot get a smooth disparity map (it has too much noise). By comparison, our method produces smooth disparity maps and preserves geometry details better than other methods. This shows that the proposed method has good generalization performance. The success of our method can be attributed to the fact that the convolutional filter bank learned by the TCSR model has a strong representation capability for geometric features of stereo images. It can also be seen from Fig. 4 that the proposed method performs better than other methods on the low texture, repetitive pattern and discontinuity regions. [2], [4], [8] and [6] not only do not produce smooth disparity maps on the low texture and discontinuity regions, but also produce a lot of noise. The proposed method performs best on the repetitive pattern. The proposed method not only produces a smooth disparity map, but also well describes the grid gaps on the repetitive pattern. [4] and [6] produce disparity maps with a lot of noise, [8] and [2] produce smooth disparity maps, but they cannot well describe the grid gaps on the repetitive pattern. In addition, the proposed method also performs better than other methods on plain color image (the last column of Fig. 4). The proposed method produces smooth disparity maps, while [2], [4], [8] and [6] produce a lot of noise on plain color image. It should be pointed out here that after training on the KITTI data set, it was directly tested on the monocular video clip without any parameter adjustment and training.

E. COMPARISON WITH OTHER STATE-OF-THE-ART METHODS

In this section, the proposed method is compared with other methods (including unsupervised methods and several

TABLE 5. Comparison of our method and some selected state-of-the-art supervised methods and existing unsupervised methods on Middlebury 2014 test dataset.

Supervised Methods	Avgerr-All [a](%)		Rms-All [b](%)		Environment	Runtime
	training set	test set	training set	test set		
CBMV [14]	11.5	14.4	34.9	46.9	Nvidia GTX TitanX (CUDA, Python, C/C++)	1001 s
JMR [54]	9.57	15.7	32.0	49.0	NVidia Titan X (C++/CUDA)	4.46 s
MC-CNN-arct [18]	11.8	17.9	36.6	55.0	Nvidia GTX Titan (CUDA, Lua/Torch7)	150 s
MC-CNN-fst [18]	12.8	19.3	37.5	55.7	Nvidia GTX Titan (CUDA, Lua/Torch7)	1.69 s
MC-CNN-WS [15]	13.7	19.9	38.3	56.6	Nvidia (GPU, Lua/Torch)	3.19 s
LW-CNN [12]	10.9	19.3	35.3	58.8	Nvidia Geforce GTX Titan X (Torch)	314 s
DSGCA [46]	18.7	26.9	45.7	66.6	NVIDIA GeForce GTX 1080 (GPU, MATLAB)	10.2 s
RBES-GC [47]	-	10.7	-	-	Intel core i5 @3.0 GHz	3.9 s
UpBIU (NSP) [55]	-	-	-	36.1	NVIDIA GeForce GTX Titan X (CUDA, c++/MATLAB)	492 s
Chen et al. [48]	-	20.0	-	-	NVIDIA Titan X (CUDA, c++/Torch7)	80 s
Unsupervised Methods	**Avgerr-All (%)**		**Rms-All (%)**			
	training set	test set	training set	test set		
MDP [56]	10.8	13.6	32.6	43.4	Nvidia GTX Titan X (Python)	79.2 s
LPS [57]	12.8	19.7	30.0	44.7	4 cores i7 @3.6GHz (c/c++)	9.52 s
MBM [49]	10.1	12.9	27.1	37.5	Nvidia GTX TITAN X (GPU)	**0.01** s
SGBMP [58]	11.2	11.4	26.7	30.8	Nvidia GTX TITAN X (Python/c++)	9.12 s
SM-AWP [59]	16.0	35.6	39.9	70.5	i7 core @2.7GHz (MATLAB)	3.3 s
MTS [53]	27.6	38.5	49.2	68	4 i7 cores @3.4GHz (c++)	2.6 s
LAMC_DSM [60]	14.6	23.1	38.4	60.1	1 core Intel Xeon @2.5 GHz (MATLAB)	704 s
Ours	**8.09**	**10.3**	**22.6**	**30.1**	1 core Intel Xeon @2.3GHz (MATLAB, c++)	49.3 s

[a] **Average error in all pixels**.
[b] **Root mean square error in all pixels. Best results are shown in bold**.

state-of-the-art supervised methods) on Middlebury 2014 data set and KITTI data set. The results are shown in Table 5 and Table 4, respectively. It can be seen from the table that our method achieves the smallest error among all unsupervised methods and supervised methods in Table 5 and Table 4.

Here, Out-noc, Avg-all, D1-bg, and D1-all are used as evaluation metrics for different methods, in which ''Out-noc'' is percentage of erroneous pixels in non-occluded, ''Avg-all'' is average disparity (end-point error) in total areas, ''D1-bg'' is percentage of outliers averaged only over background regions in first frame, and ''D1-all'' is percentage of outliers averaged over all ground truth pixels in first frame, respectively.

In terms of runtime, it can be observed from Table 5 that most of the other methods use GPU/CUDA accelerations, except for our method, LAMC_DSM [60] and LPS [57]. The running time of our method is significantly lower than that of LAMC_DSM, but higher than that of LPS. Besides, we achieve a lower computational expense, even when it is compared with supervised learning methods CBMV [14] and LW-CNN [12] which work on Nvidia GTX Titan with CUDA acceleration and unsupervised learning method MDP [56] which works on Nvidia GTX Titan X. Our method is done on the CPU and an accelerated version of the proposed method on the GPU will be considered in the future.

IV. CONCLUSION

This paper introduces the two-branch technique into the convolutional sparse representation first time in the paper and builds the TCSR model. An efficient iterative algorithm using the augmented Lagranian multiplier framework is provided to solve this tri-convex problem, and the convergence of the algorithm is discussed theoretically. Based on TCSR model, a new unsupervised stereo matching cost (USMC) is proposed in this paper. Experimental results on four popular data sets and one monocular video clip demonstrate that the USMC has higher accuracy and good generalization performance.

We would like to mention that the accuracy of proposed method is lower than those top supervised learning method. In the future, we plan to further improve the matching accuracy by adopting a multi-layer TCSR model. GPU version of the proposed method will also be considered.

REFERENCES

[1] X. Yang, X. Chen, and J. Xi, ''Block based dense stereo matching using adaptive cost aggregation and limited disparity estimation,'' in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (IMTC)*, May 2017, pp. 1–6.
[2] D. Scharstein and R. Szeliski, ''A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,'' *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, Apr. 2002.
[3] R. Ben-Ari and N. Sochen, ''A geometric approach for regularization of the data term in stereo-vision,'' *J. Math. Imag. Vis.*, vol. 31, no. 1, pp. 17–33, May 2008.
[4] J.-H. Mun and Y.-S. Ho, ''Guided image filtering based disparity range control in stereo vision,'' *Electron. Imag.*, vol. 2017, no. 5, pp. 130–136, Jan. 2017.
[5] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, ''Real-time correlation-based stereo vision with reduced border errors,'' *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 229–246, 2002.
[6] Y. Zhan, Y. Gu, K. Huang, C. Zhang, and K. Hu, ''Accurate image-guided stereo matching with efficient matching cost and disparity refinement,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1632–1645, Sep. 2016.
[7] R. Zabih and J. Woodfill, ''Non-parametric local transforms for computing visual correspondence,'' in *Proc. Eur. Conf. Comput. Vis.*, 1994, pp. 151–158.
[8] K. Yamaguchi, D. McAllester, and R. Urtasun, ''Efficient joint segmentation, occlusion labeling, stereo and flow estimation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2014, pp. 1–16.

[9] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 504–511, Feb. 2013.

[10] S. Hermann and T. Vaudrey, "The gradient—A powerful and robust cost function for stereo matching," in *Proc. 25th Int. Conf. Image Vis. Comput.*, Nov. 2010, pp. 467–474.

[11] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5695–5703.

[12] H. Park and K. M. Lee, "Look wider to match image patches with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1788–1792, Dec. 2017.

[13] M. El-Khamy, H. Ren, X. Du, and J. Lee, "Multitask deep neural networks for tele-wide stereo matching," *IEEE Access*, vol. 8, pp. 184383–184398, Oct. 2020.

[14] K. Batsos, C. Cai, and P. Mordohai, "CBMV: A coalesced bidirectional matching volume for disparity estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2060–2069, doi: 10.1109/CVPR.2018.00220.

[15] S. Tulyakov, A. Ivanov, and F. Fleuret, "Weakly supervised learning of deep metrics for stereo reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1348–1357.

[16] N. Ma, Y. Men, C. Men, and X. Li, "Segmentation-based stereo matching using combinatorial similarity measurement and adaptive support region," *Optik*, vol. 137, pp. 124–134, May 2017.

[17] A. Ahmadi and I. Patras, "Unsupervised convolutional neural networks for motion estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1629–1633.

[18] J. Žbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2287–2318, Jan. 2016.

[19] P. Brandao, E. Mazomenos, and D. Stoyanov, "Widening siamese architectures for stereo matching," *Pattern Recognit. Lett.*, vol. 120, pp. 75–81, Apr. 2019.

[20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[21] G. Huang, Y. Gong, Q. Xu, K. Wattanachote, K. Zeng, and X. Luo, "A convolutional attention residual network for stereo matching," *IEEE Access*, vol. 8, pp. 50828–50842, Mar. 2020.

[22] X. Yang, L. He, Y. Zhao, H. Sang, Z. L. Yang, and X. J. Cheng, "Multi-attention network for stereo matching," *IEEE Access*, vol. 8, pp. 113371–113382, Jun. 2020.

[23] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, "A deep visual correspondence embedding model for stereo matching costs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 972–980.

[24] A. Li and Z. Yuan, "Occlusion aware stereo matching via cooperative unsupervised learning," in *Proc. Asian Conf. Comput. Vis.*, Cham, Switzerland, May 2018, pp. 197–213.

[25] C. Zhou, H. Zhang, X. Shen, and J. Jia, "Unsupervised learning of stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1576–1584.

[26] W. Zheng, X. Zhu, G. Wen, Y. Zhu, H. Yu, and J. Gan, "Unsupervised feature selection by self-paced learning regularization," *Pattern Recognit. Lett.*, vol. 132, pp. 4–11, Apr. 2020.

[27] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, "Learning for disparity estimation through feature constancy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2811–2820.

[28] J. Pang, W. Sun, C. Yang, J. Ren, R. Xiao, J. Zeng, and L. Lin, "Zoom and learn: Generalizing deep stereo matching to novel domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2070–2079.

[29] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.

[30] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu, "AdaDepth: Unsupervised content congruent adaptation for depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2656–2665.

[31] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, "Learning monocular depth by distilling cross-domain stereo networks," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 484–500.

[32] Q. Yao, G. Luo, and Y. Zhu, "Depth estimation for outdoor image using couple dictionary learning and region detection," in *Proc. IEEE Vis. Commun. Image Process.*, Dec. 2017, pp. 1–4.

[33] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[34] S. Ravishankar, R. R. Nadakuditi, and J. A. Fessler, "Efficient sum of outer products dictionary learning (SOUP-DIL) and its application to inverse problems," *IEEE Trans. Comput. Imag.*, vol. 3, no. 4, pp. 694–709, Dec. 2017.

[35] J. Yin, H. Zhu, D. Yuan, and T. Xue, "Sparse representation over discriminative dictionary for stereo matching," *Pattern Recognit.*, vol. 71, pp. 278–289, Nov. 2017.

[36] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 391–398.

[37] A. Bibi and B. Ghanem, "High order tensor formulation for convolutional sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1790–1798.

[38] F. Heide, W. Heidrich, and G. Wetzstein, "Fast and flexible convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5135–5143.

[39] M. Šorel and F. Šroubek, "Fast convolutional sparse coding using matrix inversion lemma," *Digit. Signal Process.*, vol. 55, pp. 44–51, Aug. 2016.

[40] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. German Conf. Pattern Recognit.*, Oct. 2014, pp. 31–42.

[41] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.

[42] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.

[43] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[44] Q. Long, Q. Xie, S. Mita, H. Tehrani, K. Ishimaru, and C. Guo, "Real-time dense disparity estimation based on multi-path viterbi for intelligent vehicle applications," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.

[45] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[46] Williem and I. K. Park, "Deep self-guided cost aggregation for stereo matching," *Pattern Recognit. Lett.*, vol. 112, pp. 168–175, Sep. 2018.

[47] Y. Fu, W. Chen, K. Lai, Y. Zhou, and J. Tang, "Rank-based encoding features for stereo matching," *IEEE MultimediaMag.*, vol. 26, no. 4, pp. 28–42, Oct. 2019.

[48] L. Chen, L. Fan, J. Chen, D. Cao, and F. Wang, "A full density stereo matching system based on the combination of CNNs and slanted-planes," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 2, pp. 397–408, Feb. 2020.

[49] Q. Chang and T. Maruyama, "Real-time stereo vision system: A multi-block matching on GPU," *IEEE Access*, vol. 6, pp. 42030–42046, Jul. 2018.

[50] J. J. Yu, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 3–10.

[51] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proc. 10th Asian Conf. Comput. Vis. Comput. Vis. (ACCV)*, Nov. 2010, pp. 25–38.

[52] H. Li, L. Chen, and F. Li, "An efficient dense stereo matching method for planetary rover," *IEEE Access*, vol. 7, pp. 48551–48564, 2019.

[53] R. Brandt, N. Strisciuglio, N. Petkov, and M. H. F. Wilkinson, "Efficient binocular stereo correspondence matching with 1-D max-trees," *Pattern Recognit. Lett.*, vol. 135, pp. 402–408, Jul. 2020.

[54] P. Knobelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock, "End-to-end training of hybrid CNN-CRF models for stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2339–2348.

[55] H. Li, L. Chen, and F. Li, "Efficient confidence-based hierarchical stereo disparity upsampling for noisy inputs," *IEEE Access*, vol. 7, pp. 48551–48564, 2019.

[56] A. Li, D. Chen, Y. Liu, and Z. Yuan, "Coordinating multiple disparity proposals for stereo computation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4022–4030.

[57] S. N. Sinha, D. Scharstein, and R. Szeliski, "Efficient high-resolution stereo matching using local plane sweeps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1582–1589.
[58] Y. Hu, W. Zhen, and S. Scherer, "Deep-learning assisted high-resolution binocular stereo depth reconstruction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 467–474.
[59] S. S. A. Razak, M. A. Othman, and A. F. Kadmin, "The effect of adaptive weighted bilateral filter on stereo matching algorithm," *IJEAT*, vol. 8, no. 3, pp. 2249–8958, 2019.
[60] C. Stentoumis, L. Grammatikopoulos, I. Kalisperakis, and G. Karras, "On accurate dense stereo-matching using a local adaptive multi-cost approach," *ISPRS J. Photogramm. Remote Sens.*, vol. 91, pp. 29–49, May 2014.

CHUNBO CHENG received the B.Sc. degree in mathematics and applied mathematics from Yangtze University, Jingzhou, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan, China. His research interests include pattern recognition, machine learning, and computer vision.

HONG LI (Member, IEEE) received the M.Sc. degree in mathematics and the Ph.D. degree in pattern recognition and intelligence control from the Huazhong University of Science and Technology, Wuhan, China, in 1986 and 1999, respectively. She is currently a Professor with the School of Mathematics and Statistics, Huazhong University of Science and Technology. Her research interests include approximation theory, wavelet analysis, learning theory, neural networks, signal processing, and pattern recognition.

LIMING ZHANG (Member, IEEE) received the M.S. degree in signal processing from the Nanjing University of Science and Technology, China, and the Ph.D. degree in image processing from the University of New England, Australia. She is currently an Assistant Professor with the Faculty of Science and Technology, University of Macau, China. Her research interests include signal processing, image processing, and computer vision.